

## Acyclic non-linear graph의 빠른 신뢰도 계산방법에 관한 연구

### A Study on rapid evaluation of reliability for acyclic non-linear graph

이 광 원\* · 이 현 규\*

Kwang-Won Rhie · Hyun-Kyu Lee

(1998년 2월 11일 접수, 1998년 5월 2일 채택)

#### ABSTRACT

FTA is the most commonly used method among quantitative safety assessment. In case that the observing system become larger, a lot of terms should be calculated to accomplish FTA through complicated process. Many methods have been tried to reduce time, one of tries is How to calculate the reliability using graph theory after changing FT to graph. This paper suggests an algorithm that can calculate more rapidly reliability and cutset of system expressed by non-linear graph as like as FTA or CCA.

#### 1. 서 론

##### 1.1 연구의 배경 및 목적

Network이나 FTA와 같은 logic tree를 graph로 변환시키는 것은 예로부터 많이 연구되었다<sup>[2]</sup>. 이때 변환된 graph를 통해 신뢰도를 계산하는 데는 보통 Inclusion Exclusion Method, Sum of Disjoint Products Method, Pivotal Decomposition or Factoring 등의 3가지 방법이 있다<sup>[3~5]</sup>.

Graph나 network에 나타나는 선이 실제 부품과 1:1로 대응되는 것을 linear graph라 하며 이

에 대한 많은 연구가 발표되었다<sup>[6~13]</sup>.

그러나 우리가 흔히 사용하는 정량적 안전성 평가인 FTA나 공통원인고장분석등에서는 하나의 부품이 여러번 나타나게 된다. 본 논문에서는 이런 non-linear 형태의 graph에 있어서 빠르게 시스템의 신뢰도와 cutset들을 산출할 수 있는 알고리즘을 제시하고자 한다.

##### 1.2 이론적 배경

본 연구에서는 cycle을 포함치 않고 한 개의 선이 여러 곳에서 나타날 수 있는 acyclic non-linear graph  $G=(V, E)$ 를 관찰한다.

\* 호서대학교 안전공학부

이 그래프는 절점들의 집합  $V$ 와 선들의 집합  $E$ 로 구성되며 절점은 완벽하고 선들은 각각의 신뢰도를 갖는 것으로 가정한다<sup>14,15)</sup>.

### 1.2.1 Formation, Domination

관찰하는 어떤 acyclic non-linear graph를  $G$ 라 하고 임의의 부분 그래프를  $G_a$ 라 할 때  $G$ 의 선집합을  $E = \{e_1, e_2, e_3, \dots, e_b\}$ 라 하며 임의의 선의 부분집합  $E_a (\subseteq E)$ 는  $G_a$ 의 선집합이라 하자.

선집합  $E$ 에 포함되는 임의의 선들을 원소로 하는 임의의 집합의 family를  $M(G) = \{M_1, M_2, \dots, M_n\}$ 라 할 때  $M(G)$ 의 어떤 부분집합  $M_i = \{M_i, M_j, \dots, M_l\}$ 가

$$\bigcup_{j=1}^l M_j = E_a \quad \dots \quad (1)$$

인 경우  $G_a$ 의 formation이라 하며  $M_i$ 에 포함된  $M_j$ 의 수가 짝수일 경우 짝수 formation, 홀수인 경우 홀수 formation이라고 한다.

$N_e$ 와  $N_o$ 가 각각 짝수와 홀수 formation의 수를 나타낸다고 할 때  $M(G)$ 를 기초로 하는 부분 그래프  $G_a$ 의 domination은  $d(G_a, M(G))$ 로 표시하며

$$d(G_a, M(G)) = N_o - N_e. \quad \dots \quad (2)$$

로 정의된다.

본 연구에서는  $M(G)$ 가 m-cutset들의 family  $C(G) = \{C_1, C_2, \dots, C_m\}$ 인 경우를 관찰하게 되며 이 경우 식(2)는

$$d(G_a, C(G)) = N_o - N_e \quad \dots \quad (3)$$

이 된다.

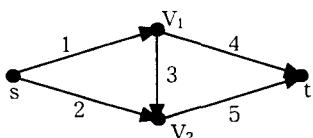


Fig. 1 Acyclic linear graph with bridge structure

### 【예제 1】

Fig. 1에 표현된 그래프  $G$ 는 선이 5개, 내부절점이 2개인 bridge구조의 acyclic linear graph이다. 이들 m-cutset의 family를  $C(G) = \{C_1, C_2, C_3, C_4\}$ 라 할 때 이들 cutset들로 graph  $G$ 에 포함되는 선들의 집합  $\{1, 2, 3, 4, 5\}$ 을 만드는

cutset들의 집합, 즉  $G$ 의 formation은  $\{C_1, C_2, C_4\}, \{C_2, C_3, C_4\}, \{C_3, C_4\}, \{C_1, C_3, C_4\}, \{C_1, C_2, C_3, C_4\}$  등 5개가 존재한다.

이때  $\{C_1, C_2, C_4\}, \{C_2, C_3, C_4\}, \{C_1, C_3, C_4\}$  등 3개는 홀수개의 m-cutset들로 구성된 홀수 formation이며  $\{C_3, C_4\}, \{C_1, C_2, C_3, C_4\}$ 는 짝수 formation이다. 결국 그래프  $G$ 에 대한 domination은

$$d(G, C(G)) = N_o - N_e = 3 - 2 = 1$$

이며, 이는 poincaré식에서  $\{1, 2, 3, 4, 5\}$ 를 갖는 항이 5번 출현되며 부호에 의해 4개항은 상쇄되고 결국 '+' 부호를 갖는 1개항만이 남게 된다는 것을 의미한다.

### 1.2.2 Indegree, Outdegree, Incut, Outcut

그래프 상에서 어떤 절점  $V_i$ 로 들어오는 선의 수를 Indegree, 나가는 선의 수를 Outdegree라고 하며 또 절점  $V_i$ 로 들어오는 선의 집합을 Incut( $V_i$ ) 또는 간단히  $I_i$ , 절점  $V_i$ 에서 나가는 선의 집합을 Outcut( $V_i$ ) 또는 간략히  $O_i$ 라 표시한다.

#### 【예제 2】

위의 Fig. 1에서 각 절점의 Indegree와 Outdegree, Incut과 Outcut은 Table 1과 같다.

Table 1 Indegree, Outdegree, Incut, Outcut of Fig. 1

|       | Indegree | Outdegree | Incut ( $I_i$ ) | Outcut ( $O_i$ ) |
|-------|----------|-----------|-----------------|------------------|
| $V_1$ | 1        | 2         | {1}             | {3, 4}           |
| $V_2$ | 2        | 1         | {2, 3}          | {5}              |

## 2. Acyclic non-linear graph의 신뢰도 계산

### 2.1 graph $G$ 의 신뢰도

지금까지는 일반적으로 어떤 시스템의 신뢰도나 고장도를 구하기 위하여 먼저 그 시스템의 m-pathset, 또는 m-cutset을 구하고 그것을 바탕으로 하여 Inclusion-Exclusion식(Poincaré식)에 대입하여 계산하게 된다. 이때 cutset의 수를 m개라 했을 때 일반식은

$$Q(G) = q \left[ \bigcup_{i=1}^m A_i \right] = \sum_{i=1}^m q(A_i) - \sum_{i=1} \sum_{j < i} q(A_i A_j) +$$

$$\dots + (-1)^{m-1} q(A_1 A_2 \dots A_m) \dots \quad (4)$$

with  $A_i$ ;  $i$ 번째 cutset에 포함되는 모든 부품이  
고장나는 사건

$q(A_i)$ ; 사건  $A_i$ 가 일어날 확률

로 표현되고 계산되어야 할 항의 수는  $2^m - 1$ 개가 된다.

보통의 경우 위식에서 계산되어지는 항들은 같은 선들을 포함하며 부호가 반대인 것들이 많이 출현하게 되며 서로 소거되어 진다. domination의 정의에 의하여 위식은 다음과 같이 쓸 수 있다.

$$Q(G) = \sum_{G_A} d(G_A, C(G)) \cdot q(G_A) \dots (5)$$

with  $q(G_A)$ :  $G_A$ 에 포함되는 모든 선들이 고장 날 확률

식(5)는 항의 수가 최대  $2^b$ 개(즉 그래프 G가 포함하는 선들에 대한 모든 부분집합)가 되며 보통의 경우 cutset의 수 m은 b보다 훨씬 크므로 식(5)의 항수는 식(4)의 항수에 비하여 훨씬 적다고 할 수 있다.

### 【예제 3】

Fig. 1에서 Poincaré식을 이용하여 고장도를 계산한다고 했을 때 구해야 되는 항은

$$\begin{aligned}
Q(G) = & q(1,2) + q(4,5) + q(1,5) + q(2,3,4) \\
& - q(1,2,4,5) - q(1,2,5) - q(1,2,3,4) \\
& - q(1,4,5) - q(2,3,4,5) - q(1,2,3,4,5) \\
& + q(1,2,4,5) + q(1,2,3,4,5) + q(1,2,3,4,5) \\
& + q(1,2,3,4,5) - q(1,2,3,4,5)
\end{aligned}$$

와 같이 총  $2^4$ –1개, 즉 15개의 항을 계산하여야 한다. 하지만 domination이론을 이용하게 되면 여기서는  $2^5$ =32개의 항, 즉, { }, {1}, {2}, …, {1, 2, 3}, {1, 2, 4}, …, {1, 2, 3, 4, 5} 등 각 선들의 조합을 구해 각각의 domination을 계산해야 되므로 본 예제에서는 Poincaré식에 비해서 많은 항을 계산하게 되지만 일반적으로 선수와 절점수가 증가하게 될수록 cutset의 수는 선의 수보다 훨씬 많아지므로 계산항의 수에 있어서 상당한 이득을 볼 수 있다는 것을 쉽게 알 수 있다.

이때 graph G에서 여러 번 중복되어 나타나는 선들을 각기 다른 성분으로 인식하여 만든 linear graph를  $G'$ 로 표시한 경우 신뢰도 관계

는 다음과 같음이 1)에서 증명되었다.

$$Q(G) = \sum_{G_A} d(G_A^L, C(G^L)) \cdot q(G_A) \dots (6)$$

with  $G_A^L$ : linear graph  $G^L$ 's subgraph

$C(G')$ :  $G^L$ 의 m-cutset의 family  
식(6)을 사용하기 위하여는 cutset family  $C(G)$ 를 기초로 한 임의의 subgraph에 대한 domination값을 계산하여야 한다.

문헌 12)에서는 이들을 계산할 수 있는 몇 가지 식을 제시하고 있다. 즐

① 그래프  $G^L$ 에서 어떤 내부 절점들의 집합 A에서 Incut를 그리고 또 다른 내부 절점들의 집합 B(단  $A \cap B = \emptyset$ )에서 Outcut를 제거할 때 얻어지는 subgraph  $G - I_A - O_B$ 들만이 domination 값이 0이 아니다.

② 임의의 subgraph  $G_A^L$

$$\begin{aligned} G_A^L &= G^L - I_{A_1}^L - O_{B_1}^L \\ &= G^L - I_{A_2}^L - O_{B_2}^L = \dots \end{aligned}$$

로 표시되고  $n$ 을 그래프  $G^L$ 의 내부 절점수라 할 때

$m_e$ :  $n - |A_i| - |B_i|$ 가 짝수인  
 $G^L - I^L_{A_i} - O^L_{B_i}$ 의 수  
 $m_o$ :  $n - |A_i| - |B_i|$ 가 홀수인  
 $G^L - I^L_{A_i} - O^L_{B_i}$ 의 수

이라면 cutset family  $C(G)$ 를 기초로 한  $G_A^L$ 의 domination값은 다음과 같다.

$$d(G_A^L, C(G^L)) \equiv m_a - m_c \quad \dots \dots \dots (7)$$

### [예제 4]

Fig. 1에서 선집합{1, 4, 5}를 갖는 임의의 부분 그래프  $G_k$ 를 관찰하자. 이 부분 그래프는 그래프  $G$ 에서 선들 2와 3을 제거한 것이며 이들은  $V_1$ 의 Incut으로만 제거가 가능하다. 즉  $G_k = G - I_1$ 으로만 표현이 가능하며 결국  $m_e = 0$ ,  $m_o = 1$ 이므로  $d(G_k, C(G)) = 0 - 1 = -1$ 이 된다.

식(6)에서 domination값이 0인 subgraph들은 신뢰도 계산에 불필요하고 domination값이 0이 아닐 수 있는 subgraph들의 관찰이 중요하다. 이들의 family  $SIO(G^l)$ 는 다음과 같이 정의되며 총수는 3<sup>n</sup>개이다.

$$S(\mathcal{O}(G^L)) = \{ G^L = I_A^L = O_B^L \mid \forall A \subseteq N \}$$



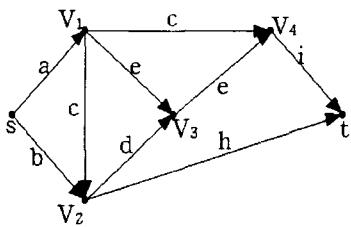
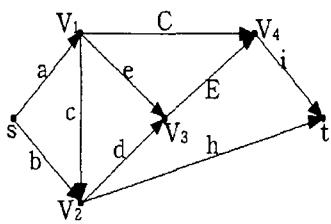


Fig. 2 Acyclic non-linear graph G

Fig. 3 acyclic linear graph G<sup>L</sup>

{V<sub>1</sub>, V<sub>2</sub>, V<sub>3</sub>, V<sub>4</sub>} 등 12개이며 이는 Fig. 4에 도시되었다. 이에 따른 SIO(G<sup>L</sup>)와 SIO(G)를 Fig. 5에 도시하였다.

이 그래프의 고장도 Q(G)는 식(10)에 의하여 다음과 같이 쓰여진다

$$\begin{aligned}
 Q(G) = & q(ab) + q(bce) - q(abce) + q(adh) \\
 & - q(abdh) + q(cdeh) - q(acdeh) \\
 & - q(bcdeh) + q(abcdeh) + q(bce) \\
 & - q(abce) - q(bce) + q(abce) + q(bcei) \\
 & - q(abcei) - q(bcei) + q(abcei) + q(aeh) \\
 & - q(abe) - q(adeh) + q(abdeh) \\
 & + q(ceh) - q(aceh) - q(bceh) - q(cdeh) \\
 & + q(abceh) + q(acdeh) + q(bcdeh) \\
 & - q(abcdeh) + q(deh) - q(adehi) \\
 & - q(bcdehi) - q(dceh) + q(abcddeh) \\
 & + q(adcehi) + q(bcdehi) - q(abcdehi) \\
 & + q(bci) - q(abci) - q(bcei) - q(bcei) \\
 & + q(abcei) + q(abcei) + q(bcei) \\
 & - q(abcei) + q(ahi) - q(abhi) - q(adhi) \\
 & - q(aehi) + q(abdh) + q(abehi) \\
 & + q(adehi) - q(abdehi) + q(hi) - q(ahi) \\
 & - q(bchi) - q(adehi) - q(cehi) + q(abchi) \\
 & + q(adehi) + q(acehi) + q(bcdehi) \\
 & + q(abcehi) + q(adehi) - q(abcdehi) \\
 & - q(abcei) - q(acdei) - q(bcdehi) \\
 & + q(abcdehi)
 \end{aligned}$$

다시 고장도 Q(G)는 식(13)에 의하여 다음과 같이 쓰여진다.

$$\begin{aligned}
 Q(G) = & (ab) + q(bce)(1 - q(abce)) \\
 & + q(adh)(1 - q(abdh)) + q(cdeh)(1 - q(a)) \\
 & - q(bc) + q(abc)) + q(bce)(1 - q(a)) \\
 & - q(e) + q(ae)) + q(bcei)(1 - q(a) - q(c)) \\
 & + q(ac)) + q(aeh)(1 - q(b) - q(d)) \\
 & + q(bd)) + q(ceh)(1 - q(a) - q(bc)) \\
 & - q(de) + q(ade) + q(bcde) - q(abcde)) \\
 & + q(dehi)(1 - q(a) - q(c) + q(ac)) \\
 & + q(bci)(1 - q(a) - q(e) + q(ae)) \\
 & + q(ahi)(-q(b) - q(d) + q(bd) + q(be)) \\
 & + q(de) - q(bde)) + q(hi)(1 - q(a)) \\
 & (1 - q(bc) + q(bce) - q(de) + q(cde) \\
 & - q(ce))
 \end{aligned}$$

위식은 다시 식(15)를 사용하여 다음과 같이 표현된다.

$$\begin{aligned}
 Q(G) = & q(ab) + q(bce)(1 - q(a)) + q(adh) \\
 & (1 - q(b)) + q(cdeh)(1 - q(a))(1 - q(bc)) \\
 & + q(bce)(1 - q(a))(1 - q(e)) + q(bcei) \\
 & (1 - q(a))(1 - q(c)) + q(aeh)(1 - q(b)) \\
 & (1 - q(d)) + q(ceh)(1 - q(a))(1 - q(bc)) \\
 & (1 - q(de)) + q(dehi)(1 - q(a))(1 - q(c)) \\
 & + q(bci)(1 - q(a))(1 - q(e)) + q(ahi) \\
 & (1 - q(b))(1 - q(d))(1 - q(e)) + q(hi) \\
 & (1 - q(a))(1 - q(bc))(1 - q(e)) - q(de) \\
 & (1 - q(c)) - q(ce))
 \end{aligned}$$

위식에서 계산되어지는 항수는 총 37개이며 m-cutset들은 C(G)={\{a, b\}, \{b, c, e\}, \{a, d, h\}, \{a, e, h\}, \{c, e, h\}, \{b, c, i\}, \{h, i\}}로서 7개이고 Poincaré식에 의해 계산하려면 127개를 계산하여야 한다.

Table 2 No. of Term by Eq12, poincaré Eq, 7) and 12)

| case             | No. of Term by Eq. 12 | poincaré Eq.    | 7)    | 12)   |
|------------------|-----------------------|-----------------|-------|-------|
| linear graph     | $2^n$                 | $2^m - 1$       | $2^b$ | $2^n$ |
| non-linear graph | $< 3^n$               | $= 2^{2^n} - 1$ | .     | .     |

n개의 절점과 b개의 선을 갖는 graph에서 가장 복잡한 perfectgraph인 경우 m.cutset의 수는  $m=2^n$ 개이며 본 논문의 알고리즘과 poincaré식, 문헌 7)과 12)의 계산항 수는 Table 2와 같다.

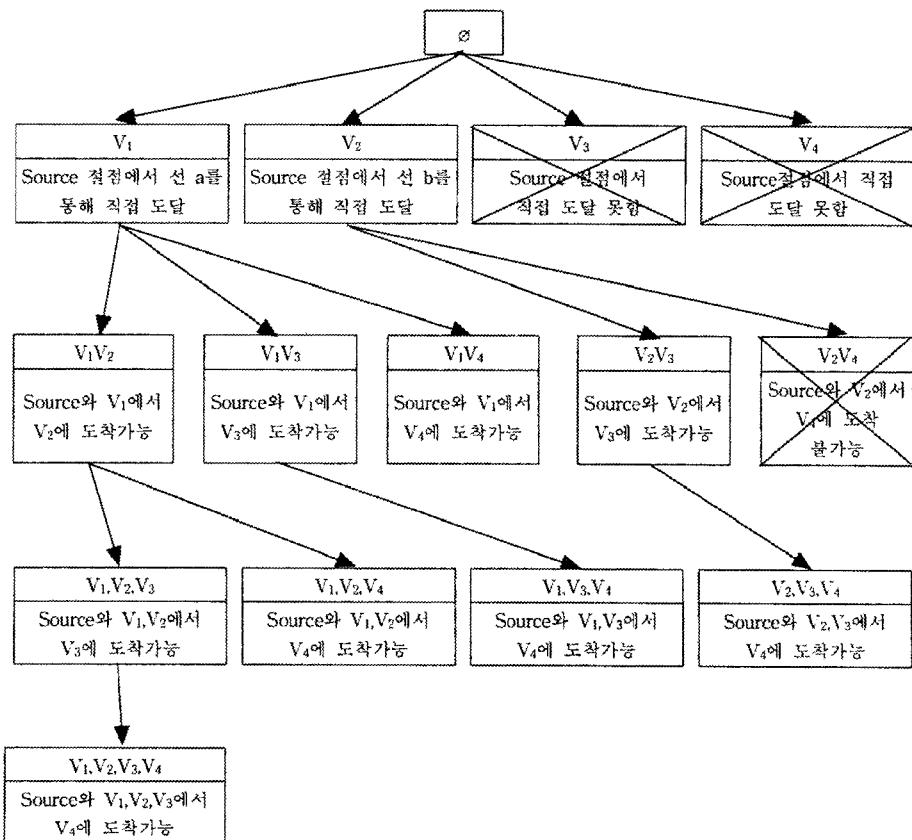


Fig. 4 Tree getting all A(G)

일반적으로  $n < b \ll m$ 임을 생각하면 본 논문의 방법이 이론적으로 훨씬 빠름을 알 수 있다.

## 2.2 알고리즘

본 프로그램은 다음과 같이 main 함수부, Recall1 함수부, Recall2 함수부의 3개의 주요 함수로 구분되어지며 그 알고리즘은 다음과 같다.

Main program

Start

Step 1: non-linear graph를 linear graph로 바꾼다.

Step 2:  $A = \emptyset$ ,  $\text{Top}^L = O_s^L$ ,  $Q(G) = q(O_s)$

Step 3:  $O_s^L$ 에 포함되는 모든 도착 절점 i에 대하여 Recall1(A, i, Top<sup>L</sup>) 호출

End

Recall1(A, i, Top<sup>L</sup>)

Start

Step 4:  $A = A \cup i$ ,  $\text{Top}^L = \text{Top}^L + q(O_s) - I_i^L$ ,  $Q = I_i^L$ ,  $B = A$

Step 5: B에 포함되는 모든 절점 i에 대하여  $Son_i^L = I_i^L - O_B^L - \text{Top}^L$ 를 구한다.

Step 6: B에 포함되는 모든 절점 i에 대하여  $Son_i^L$ 가 minimal이 아니면 제거

$B = B - i$

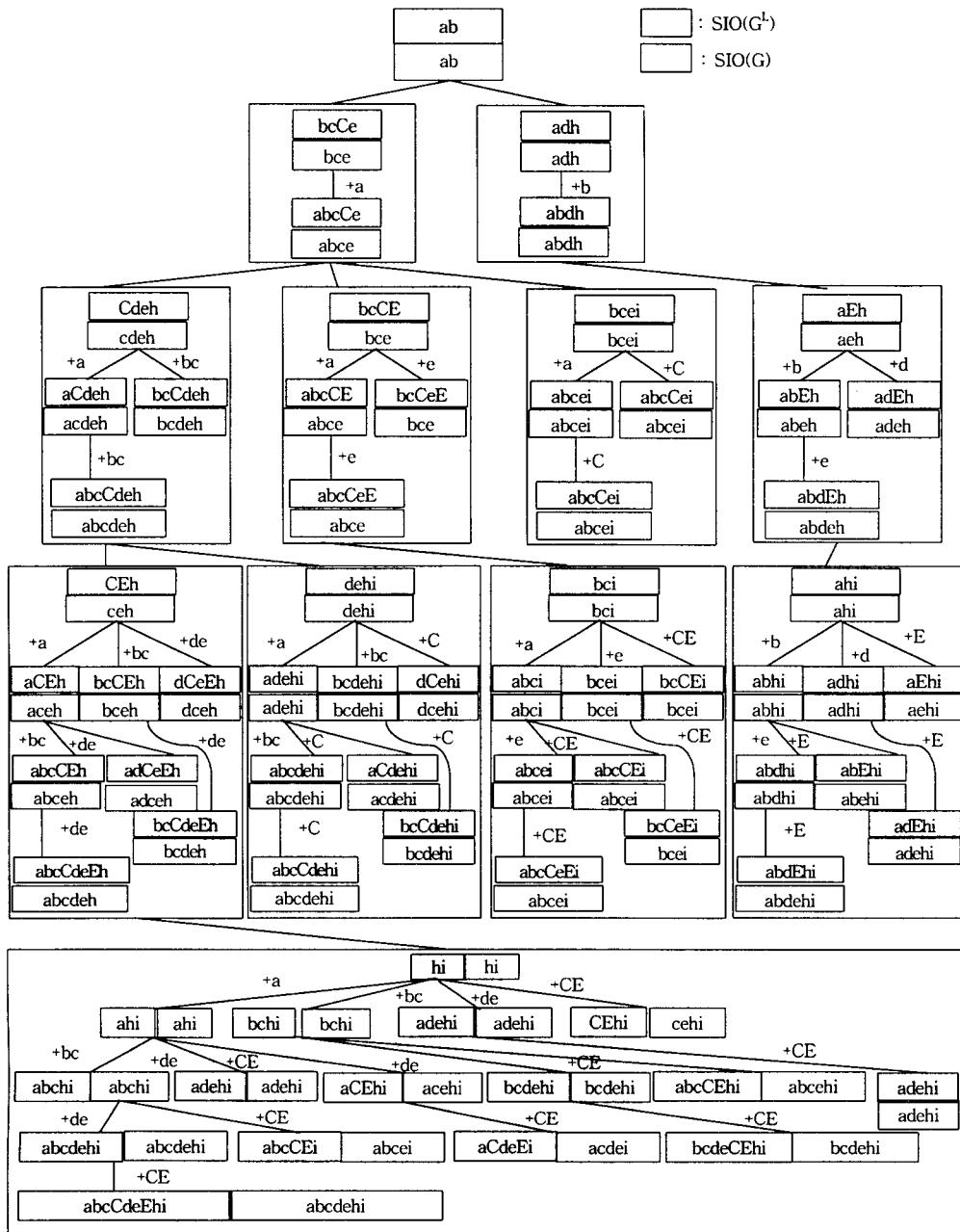
Step 7: B에 포함되는 모든 절점 i에 대하여  $Son_i^L$ 가 다른 Son과 disjoint이면

$Q = Q \cdot q(1 - q(Son_i))$

$B = B - i$

Step 8: 남은 절점 집합 B가 없으면 goto Step 11.

Step 9: 남은 절점 집합 B가  $\{b_1, b_2, \dots, b_l\}$ 인

Fig. 5 SIO(G) and SIO(G<sup>L</sup>) expressed tree form

경우 모든  $b_i$ 에 대하여

$$Top_i^L = Son_i$$

$$B_i = b_{i+1}, b_{i+2}, \dots, b_i$$

$$Q_i = q(Son_i)$$

$$Recall2(B_i, Top_i^L, c)$$

Step 11:  $Top_i^L$ 에 포함되는 선들이 도착되는 내

부 절점들 k에 대하여(단,  $k \in A$ )

$$Recall1(A, k, Top_i^L)$$

STOP  
 Recall2(B, Top<sup>L</sup>, Q)  
 Start  
 Step 12: B에 포함되는 모든 절점 j에 대하여  
 $Son_j^L = I_j^L - O_j^L - Top_j^L$ 를 구한다.  
 $Son = \{Son_1, Son_2, Son_3, \dots, Son_k\}$   
 Step 13: B에 포함되는 모든 절점 i에 대하여  
 $Son_i$ 가 minimal이 아니면 제거  
 $B = B - i$   
 Step 14: B에 포함되는 모든 절점 i에 대하여  
 $Son_i$ 가 다른  $Son$ 과 disjoint이면  
 $Q = Q \cdot q(1 - q(Son_i))$   
 $B = B - i$   
 Step 15: 남은 절점 집합 B가 없으면 end.  
 Step 16: 남은 절점 집합 B가  $\{b_1, b_2, \dots, b_l\}$ 인 경우 모든  $b_i$ 에 대하여  
 $Top_i^L = Son_i$ ,  
 $B_i = b_{i+1}, b_{i+2}, \dots, b_l$   
 Step 17:  $Q_i = q(Son_i)$   
 $Recall2(B_i, Top_i^L, Q_i)$   
 Step 18:  $Q = Q(1 - Q_1 - Q_2 - \dots - Q_l)$   
 End

### 3. 결 론

본 연구에서는 FT의 graph 변환시 등에서 나타나는 cycle을 포함하지 않고 하나 이상의 선이 중복되어 나타나는 acyclic non-linear graph에 대하여 관찰하여 보았다. domination이론을 사용하여 불필요한 항들을 제거하고 꼭 계산에 필요한 항만을 구한 후, 이들을 최대한 “묶어서” 계산하는 방법을 제시하고, 또 linear일 때와 유사하게 감소시켰다. 이것을 예제를 통하여 보았을 때 계산하는 항이 현저히 감소함을 확인할 수 있었다. 본 논문에서 설명한 알고리즘을 적당한 프로그램 언어로 실현화 시키는 것과 서론부분에서 언급되었던 FTA나 공통원인 고장분석방법등과 같이 하나의 사상이 중복되어 발생되는 위험성 평가의 방법의 경우에 본 연구에서 제시한 방법과의 접목을 통하여 현저한 계산시간의 향상을 가져올 수 있으리라 예상되며, 학문적으로는 cycle을 포함한 cyclic non-linear

graph에서의 신뢰도 계산 방법 연구가 또 하나의 과제라 하겠다.

### 참 고 문 헌

- 1) 이일재, “FT의 빠른 신뢰도 계산을 위한 연구”, 호서대학교 산업안전공학과 석사학위논문, 1997.
- 2) 日本綜合安全研究所, “FTA 安全工學”, 幾電研究社, pp. 77~80, August, 1990.
- 3) M.S. Abdel-Hameed, Erhan Cinlar and Joseph Quinn, “Reliability Theory and Models”, ACADEMIC PRESS, INC, pp. 221~229, 1984.
- 4) R.E. Barlow and S. Iyer, “Computational complexity of coherent systems and the reliability polynomial”, Probability in the Engineering and Information Sciences, pp. 461~469, February, 1988.
- 5) M.O. Locks, “Recursive Disjoint Products: A Review of Three Algorithms”, IEEE Transactions on Reliability, Vol. R-31, No. 1, April, 1982.
- 6) A. Satyanarayana, “A unified Formula for Analysis of Some network Reliability Problems”, IEEE Trans. Reliability, Vol. R-31, No. 1, April, 1982.
- 7) A. Satyanarayana and A. Prabhakar, “New Topological Formula and Rapid algorithm for Reliability Analysis of Complex network”, IEEE Trans. Reliability, Vol. R-28, April, 1979.
- 8) J.A. Abraham, “An improved algorithm for network reliability”, IEEE Trans. Reliability, Vol. R-28, April, 1979.
- 9) L. Fratta and U.G. Montanari, “A Boolean algebra method for computing the terminal reliability in a communication network”, IEEE Trans. Circuit Theory, Vol. C-20, May, 1973.
- 10) K.K. Aggarwal, K.B. Misra and J.S. Gupta, “A fast algorithm for reliability

- evaluation", IEEE Trans. Reliability, Vol. R-24, April, 1975.
- 11) Kwang-won Rhie, Zur Domination und Zuverlaessigkeit linearer Graphen aufgrund ihrer Minimal Schnitte, Dissertation, Technische Universitaet in Berlin, July, 1994.
- 12) 이광원, "Domination이론을 이용한 acyclic digraph의 빠른 신뢰도 계산을 위한 연구", 한국산업안전학회지, Vol. 11, No. 1, 1996.
- 13) 이광원, 이일재, 강신재, "Domination이론에 서의 새로운 식과 이의 신뢰성 계산에 대한 적용", 한국산업안전학회지, Vol. 11, No. 1, 1996.
- 14) M. N. S. Swamy and K. Thulasiraman, "Graphs, Networks, and Algorithms", Wiley-Interscience, pp. 3~6, September, 1980.
- 15) 吳英煥, "통신망의 신뢰도 계정에 관한 연구", 인하대학교 전자공학과, pp. 1~5, August, 1993.